Region or Global? A Principle for Negative Sampling in Graph-based Recommendation

Zhen Yang, Ming Ding, Xu Zou, Jie Tang, *Fellow,IEEE,* Bin Xu, Chang Zhou, and Hongxia Yang

Abstract—Graph-based recommendation systems are blossoming recently, which models user-item interactions as a user-item graph and utilizes graph neural networks (GNNs) to learn the embeddings for users and items. A fundamental challenge of graph-based recommendation is that there only exists observed positive user-item pairs in the user-item graph. Negative sampling is a vital technique to solve the one-class problem and is widely used in many recommendation methods. However, the previous works only focus on the design of negative sampling distribution but ignore the sampled region for negative sampling. In this work, we propose *the Three-Region Principle* to guide negative sampling, which suggests that we should negatively sample more items at an intermediate region and less adjacent and distant items. In light of this principle, we present the RecNS method, which is a general negative sampling method designed with two sampling strategies: positive-assisted sampling and exposure-augmented sampling. Instead of sampling existing negative items from graph data, we merge these two strategies in embedding space to generate negative item embeddings. Extensive experiments demonstrate that RecNS method significantly outperforms all negative sampling baselines, e.g., 10.47% for PinSage, 6.02% for NGCF, and 8.20% for LightGCN in terms of Recall@20 on the Alibaba dataset.

Index Terms—Negative Sampling, The Three-Region Principle, Embedding Mergence, Graph-based Recommendation.

1 Introduction

Recent years have seen research highlights of recommendation systems, evolving from collaborative filtering (CF) to graph-based recommendation [1], [2], [3], [4], [5]. Graph-based recommendation models user-item interactions as a user-item graph and leverages graph neural networks (GNNs) [6], [7], [8] to incorporate structural information into user/item embeddings learning. Its key point is to learn the high-quality embeddings and estimate the likelihood of a user-item interaction with learned embeddings, which is widely used for online shopping, social network, and advertising. The rapid development of GNNs has been serving as the fundamental driving force behind the rising of graph-based recommendation. Notably, graph-based recommendation has exhibited the potential to be the key technology for the next-generation recommendation, facilitating web-scale applications and thus showing a promising prospect.

However, there remains a crucial challenge that only positive pairs are observed in the user-item graphs while the other items are not connected to users that be regarded as unobserved pairs. Seriously, the number of global unobserved items is usually huge and the calculation of all the unobserved pairs is impractical. Negative sampling is a vital technique to address this issue.

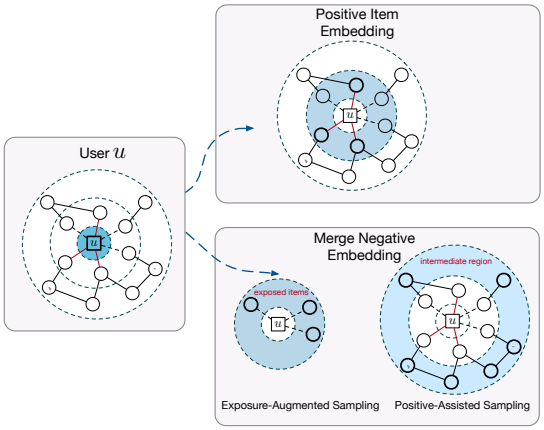


Fig. 1. An illustration of RecNS for merging positive-assisted sampling and exposure-augmented sampling in embedding space.

Negative sampling has been widely adopted in previous works [9], [10], [11], [12], the sampling strategy involves only picking a small portion of negative items from the global unobserved item region, and train models to separate those negative items from positive ones. The negative sampling strategy accelerates the training process and reduces computational complexity, making it possible for a large-scale graph-based recommendation. Besides, results in several studies [8], [13] demonstrate that the quality of negative items does affect the user/item embedding qualities and the effectiveness of the recommendation task. [14] studies the difficulty of negatives for learning useful representations and finds that just the hardest 5% negatives are both necessary and sufficient for the downstream tasks. Commonly, a classical strategy is to use a uniform distribu-

- *Zhen Yang is with the Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China. E-mail:zheny2751@gmail.com*
- *Ming Ding and Xu Zou are with the Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China. E-mail:{dm18,zoux18}@mails.tsinghua.edu.cn*
- *Jie Tang is with the Department of Computer Science and Technology, Tsinghua University, and Beijing Academy of Artifical Intelligence (BAAI), Beijing, China, 100084. E-mail: E-mail:jietang@tsinghua.edu.cn*
- *Bin Xu is with the Department of Computer Science and Technology of Tsinghua University, Beijing, China, 100084. E-mail: E-mail:xubin@tsinghua.edu.cn*
- *Chang Zhou and Hongxia Yang are DAMO Academy, Alibaba Group, Hangzhou, China, 310052. E-mail:{ericzhou.zc,yang.yhx}@alibaba-inc.com*

tion for negative sampling [4], [15]. To improve the quality of negative items, studies have attempted to design new negative sampling distributions to sample *hard* (a.k.a difficult) negative items based on the current model [16], [17], [18], in which the model would distinguish the differences between positive and negative pairs at a finer granularity. In graph-based recommendation, negative sampling strategies, such as MCNS [19] and PinSage [1], sample the negative items based on self-contrast approximation and Metropolis-Hastings, or based on their PageRank scores respectively. However, these works in graph-based recommendation only focus on designing negative sampling distribution, ignoring the choice of sampling region in the GNNs' information propagation mechanism.

Take LightGCN [4] for example, its experimental results demonstrate that the performance begins to decrease after reaching the peak point on layer 2 in most cases when the layer number increases from 1 to 4. In response to these results, the authors claim that smoothing a node's embedding with its first-order and second-order neighbors is very useful, but will suffer from over-smoothing issues when higher-order neighbors are used. Actually, the over-smoothing is inevitable when deepening the network layers but we can design a more effective negative sampling method to further improve recommendation performance. In this paper, we utilize graph structure to sample negative items in terms of structural similarities. In the user-item graph, the neighbors in smaller hop have higher chances of being related to the central node, which can improve performance by propagating information in smaller-hop neighbors. It shows that the information propagated in smaller-hop neighbors is more likely to be positive than negative for the central node. However, propagating information in the higher-hop neighbors results in performance degradation, which illustrates that these neighbors information is harmful to recommendation performance. Based on the abovementioned observations, negative items should be sampled from some specific region rather than the global unobserved region. Thus, we propose the three-region principle to sample negative items from the intermediate region, which provides a principle to answer which region should be considered as the candidate negative region. Next, we propose the negative sampling method RecNS to design a negative sampling distribution to sample negative items from the candidate negative region, which supplies a method to answer how to sample negative items. In summary, the three-region principle is a general principle to guide the selection of candidate negative item region, which leverages graph structure rather than over-smoothing to sample negatives.

Contributions. In this paper, we propose the qualitative *Three-Region Principle* to guide negative sampling, which suggests that we should negatively sample more items at an intermediate region for each user and less adjacent and distant items. With the guidance of this principle, we present an effective negative sampling strategy called RecNS to sample hard negative items (See Figure 1), which can be directly plugged into existing graph-based recommendation models, such as PinSage, NGCF, and LightGCN. To mine hard negative items for graph-based recommendation, RecNS designs two strategies: positive-assisted sampling (called RecNS-O) and exposure-augmented sampling (called RecNS-W). In positive-assisted sampling, we balance the influence between the central user and positive item on negative sampling. In exposure-augmented sampling, we incorporate exposure information into negative sampling. Finally, we merge the positive-assisted sampling and the exposure-augmented sampling in embedding space to generate the final negative item embeddings.

We conduct extensive experiments on two real-world datasets with three representative graph-based recommendation models. Experimental results demonstrate that RecNS can achieve better performance by substituting the default negative sampling strategy, such as the average gains of 10.47% for Pinsage, 6.02% for NGCF and 8.20% for LightGCN in terms of Recall@20 on the Alibaba dataset.

We summarize our key contributions as follows:

- Instead of sampling from the global unobserved item region, we propose the *Three-Region Principle* to sample negative items from the intermediate region for exploring more informative candidate negative items.
- We present a novel RecNS method that merges the positive-assisted sampling and exposure-augmented sampling in embedding space to generate the final negative item embeddings, which can be plugged into graph-based recommendation models.
- Comprehensive experimental results demonstrate that RecNS is superior to the existing negative sampling strategies.

2 FRAMEWORK

In this section, we firstly elaborate on the proposed framework SampledRec. Next, we review the overall process of SampledRec. The typical flow of the SampleRec framework is illustrated in Figure 2.

2.1 The SampledRec Framework

The proposed general graph-based recommendation framework SampledRec consists of a GNNs-based encoder E_θ to learn embeddings for items and users, a positive sampler S_p, and a negative sampler S_n to sample positive and negative items respectively for any given user (See Figure 2). The sampled user-item interactions serve as the training data for graph-based recommendation learning with stochastic gradient descent (SGD) optimizer. After training, the recommender system recommends the top K items with largest $E_\theta(u) \cdot E_\theta(v)$ for a queried user.

2.2 GNNs-based Encoders.

GNNs-based encoders play a critical role in the SampledRec, which can be briefly summarized as three modules.

Aggregation Module. We maintain an initial item and embedding matrix $\mathbf{E}_\mathcal{V} \in \mathbb{R}^{N\times d}$ and a user embedding matrix $\mathbf{E}_\mathcal{U} \in \mathbb{R}^{M\times d}$. A look-up operation is applied to form an initial embedding vector $\mathbf{e}_u \in \mathbb{R}^d$ ($\mathbf{e}_v \in \mathbb{R}^d$), where d denotes the embedding dimension. Intuitively, there are two types of aggregation operations: item and user aggregations:

$$\begin{aligned}\mathbf{h}_u &= \mathrm{Agg}_{u\leftarrow v}(\mathbf{e}_v|v \in \mathcal{S}(\mathcal{N}_u)),\\ \mathbf{h}_v &= \mathrm{Agg}_{v\leftarrow u}(\mathbf{e}_u|u \in \mathcal{S}(\mathcal{N}_v)).\end{aligned} \tag{1}$$

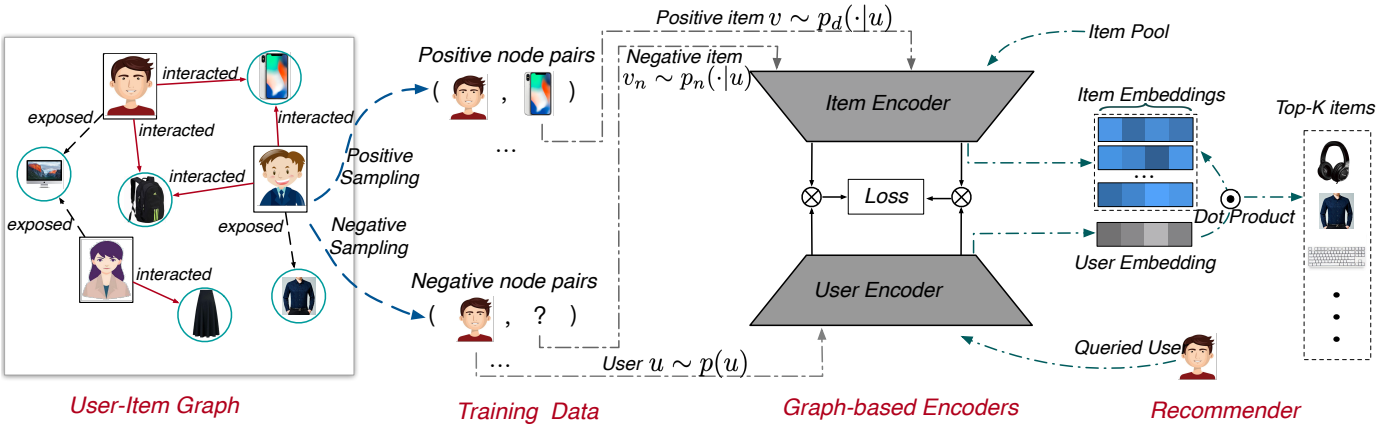


Fig. 2. An illustration of general graph-based recommendation framework SampledRec. In the user-item graph, the user and item nodes are connected to each other based on the records of interactions (such as interacted, exposed (i.e. exposed to the user but not interacted with)). In many cases, positive pairs are sampled from direct edges in the graph and negative pairs are sampled from the global unobserved item region via a pre-defined distribution, both of them composing the training data of graph-based recommendation systems.

where $\mathcal{N}_u/\mathcal{N}_v$ denotes neighbors of the central user u/item v. $\mathbf{h}_u$ and $\mathbf{h}_v$ are the aggregated embeddings for user u and item v respectively. $\text{Agg}_{u\leftarrow v}/\text{Agg}_{v\leftarrow u}$ is the user/item aggregation function. $\mathcal{S}(\cdot)$ represents neighbor sampler.

Propagation Module. To capture higher-order interactions between user and item, we stack multiple propagation layers to propagate embeddings layer by layer. Let $\mathbf{h}_u^l/\mathbf{h}_v^l$ represents user/item embedding at the l-th layer. The embeddings in $(l+1)$-th layer depends on neighbor's embeddings at l-th layer and its own embedding at l-th layer. Mathematically, the user embeddings at $(l+1)$-th layer $\mathbf{h}_u^{l+1}$ can be defined as:

$$\begin{aligned}\mathbf{h}^{l+1} &= \text{Agg}_{u\leftarrow v}(\mathbf{e}_v^l | v \in \mathcal{S}(\mathcal{N}_u)),\\ \mathbf{h}_u^{l+1} &= f(\mathbf{h}^{l+1}, \mathbf{h}_u^l).\end{aligned} \tag{2}$$

where $f(\cdot)$ is an update function. Similarity, the item embedding vector at $(l+1)$-th layer also be represented by the abovementioned propagation module.

Prediction Module. After propagating with L layers, we obtain every layer representations $\{\mathbf{h}_u^1, \cdots, \mathbf{h}_u^L\}/\{\mathbf{h}_v^1, \cdots, \mathbf{h}_v^L\}$ for the user u/item v, respectively. We utilize the representations of all layers to obtain the final user/item embeddings $\mathbf{e}_u^*/\mathbf{e}_v^*$ for prediction can be formulated as:

$$\begin{aligned}\mathbf{e}_u^* &= g(\mathbf{h}_u^1, \cdots, \mathbf{h}_u^L),\\ \mathbf{e}_v^* &= g(\mathbf{h}_v^1, \cdots, \mathbf{h}_v^L).\end{aligned} \tag{3}$$

where $g(\cdot)$ denotes a fusion function.

Finally, we use the common way, inner product, to estimate the user's preference towards the target item:

$$\hat{r}_{uv} = \mathbf{e}_u^* \cdot \mathbf{e}_v^* \tag{4}$$

2.3 Samplers

Neighbor Sampler $\mathcal{S}$. It is necessary to sample neighbors for the central node to apply GNN for large-scale graphs. GCN adopts the full neighbors for aggregation, while PinSage samples fixed-size neighbors. FastGCN [20] suggests sampling neighbors in each convolutional layer. AS-GCN [21] proposes an adaptive layer-wise neighbor sampling approach. Neighbor sampling is a vital technique to counterpoise original graph information propagation and computation efficiency.

Positive Sampler $\mathcal{S}_p$. In SampledRec, edges in the user-item graph can be assumed as positive user-item interactions. Thus, the positive user-item interactions $\mathcal{O}^+$ can be defined as $\mathcal{O}^+ = \{(u,v)|u \in \mathcal{U}, v \in \mathcal{V}\}$, where $\mathcal{U}$ indicates a set of users and $\mathcal{V}$ denotes a set of items, each pair (u,v) indicates a connected edge in the user-item graph.

Negative Sampler $\mathcal{S}_n$. In SampledRec, we conduct a negative sampler $\mathcal{S}_n$ to sample negative items, that is, $v_n \sim \mathcal{S}_n(u,v)$. Although works on negative sampling have been studied for a long time, they pay more attention to how to design negative sampling distribution p_n and ignore the selection of the negative sampling region.

2.4 Optimization

In SampledRec, we choose the hinge loss to optimize the parameters of the GNNs-based encoders, which is defined on *one-pair loss*. Here, we extend the standard hinge loss to k-pair loss, named as *the augmented hinge loss*, to enhance the performance of graph-based recommendation. For a target user u and the corresponding positive item v, we sample k negative items $v_n s$ to optimize *the augmented hinge loss*:

$$\mathcal{L} = \frac{1}{k} \sum_{\substack{(u,v)\in\mathcal{O}^+ \\ v_n\sim\mathcal{S}_n(u,v)}} [\sigma(\sum_{i=1}^{k} \mathbf{e}_u^* \cdot \mathbf{e}_{v_n^i}^*) - \sigma(k \cdot \mathbf{e}_u^* \cdot \mathbf{e}_v^*) + \gamma]_+. \tag{5}$$

where $\{v_n^1, \cdots, v_n^k\}$ denote k sampled negatives, $\sigma(\cdot)$ is the sigmoid function, $[z]_+ = max(0,z)$, γ is the pre-defined margin satisfied $\gamma > 0$, $\mathcal{O}^+$ denotes the set of positive user-item pairs, $\mathcal{S}_n(u,v)$ is the designed negative sampler.

3 The Three-Region Principle

In this section, we first introduce negative sampling problem and analyze negative sampling in graph-based recommendation. Next, we propose *the three-region principle* to guide negative sampling and give the separating criterion. Lastly, we discuss the proposed three-region principle, which is a general principle to guide negative sampling.

3.1 Negative Sampling Problem

As demonstrated in the augmented hinge loss function (5), the negative sampling plays a critical role in model training,

which allows models to learn an appropriate boundary to discriminate between positive and negative interactions. To be specific, the negative sampler $\mathcal{S}_n$ samples negatives from a designed distribution p_n, where the negatives close to positives (a.k.a hard negative items) enable the recommendation models to achieve excellent performance. Several attempts have been made to design more complicate negative sampling distribution p_n to improve the estimation of recommendation systems [16], [19], [22], [23], [24].

However, early attempts focus on seeking better negative sampling distributions to reduce estimation variance in real-world graph data. Notably, the selected region of negatives has not been fully explored. In this work, we seek the selection of regions for negative sampling.

3.2 Analysis on Negative Sampling

In this subsection, we analyze negative sampling from iterative GNNs and variance perspectives. **Iterative GNNs.** The idea of iterative GNNs is to propagate information layer by layer in the user-item graph to generate user/item embeddings. Notably, the number of propagation layers has a huge influence on performance, whose results are revealed in several papers [4], [15]. From the perspective of negative sampling, the smaller hop of neighbors (from one-hop to three-hop) indicates a positive preference, improving the performance of graph-based recommendations. Here, we use the social influence theory [25], [26] to explain why the smaller hop of neighbors can improve the performance. As well supported by the social influence theory, users in social networks (a.k.a user-user social graph) would influence each other, which leads to similar preferences. Similarly, users interacting with the same item means that these users belong to the same community and possess the same preference. Hence, the two-hop neighbors can be integrated into the central node to improve performance (a.k.a $u \leftarrow v \leftarrow u$ or $v \leftarrow u \leftarrow v$). Meanwhile, for three-hop neighbors, take the central user u_0 as an example (a.k.a $u_0 \leftarrow v_0 \leftarrow u_1 \leftarrow v_1$), the central user u_0 and its two-hop neighbor u_1 belong to the same community due to they all interacted with the same item v_0, and thus the u_1's one-hop neighbor v_1 would interacted by the central user u_0 with a high probability. However, the larger hop of neighbors (such as four-hop or even five-hop) may demonstrate a negative preference, which will degrade the recommendation performance. Thus, the region for negative sampling should be separated by the propagation mechanism in iterative GNNs. **Variance.** Recently, a study on negative sampling for graph representation learning (MCNS) [19] theoretically demonstrates that the expected risk between expected loss $\mathcal{J}(\theta^*)$ and empirical loss $\mathcal{J}(\theta_T)$ satisfies:

$$\mathbb{E}[||(\theta_T - \theta^*)_u||^2] = \frac{1}{T}\left(\frac{1}{p_d(v|u)} - 1 + \frac{1}{kp_n(v|u)} - \frac{1}{k}\right) \quad (6)$$

where $p_d(v|u)$, $p_n(v|u)$ denote the estimated positive distribution and negative distribution, respectively. T denotes the number of sampled samples, where $\{v_1, ..., v_T\}$ are sampled from estimated $p_d(v|u)$ and $\{v'_1, ..., v'_{kT}\}$ are sampled from $p_n(v|u)$. k denotes the number of negative samples for each positive user-item pair. This derivation suggests to sample negative nodes positively but sub-linearly correlated to their positive sampling distribution. i.e. $p_n(v|u) \propto p_d(v|u)^\alpha, 0 < \alpha < 1$. Hence, the order of magnitude of the expected risk only negatively related to $p_d(v|u)$, and the user-item interactions with high inner product score can be estimated more accurately.

Based on the abovementioned theories, the suggested way for negative sampling contains two steps: first, the selected region for negative sampling should be determined by iterative GNNs, namely *the three-region principle*; second, the negative sampling distribution should be satisfied with the expected risk. For the second step, we propose a negative sampling method called RecNS in Section 4.

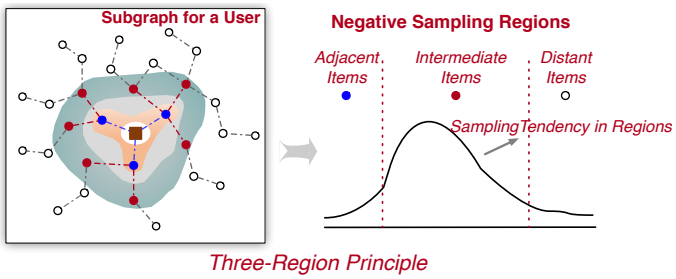


Fig. 3. The subgraph for a user contains three types of items, representing three regions for negative sampling. This paper suggests to sample negative items according to the Three-Region Principle.

3.3 The Three-Region Principle for Negative Sampling

Oriented to the property of the propagation mechanism of GNNs in graph-based recommendation task, we propose **the three-region principle** for negative sampling related to any user u, where items are divided into three categories (See Figure 3):

- **Adjacent Region.** The items in the adjacent region are usually used for propagating feature information for the central user u in graph-based recommendation. Therefore, these items represent the positive preference of the user and should not be sampled as negative ones. Besides, according to equation (6), we need to keep $p_n(v|u) > 0$ to guarantee the optimal $\mathbf{e}_u \cdot \mathbf{e}_v(p_d(v|u)) < +\infty$. In summary, these items are usually not taken into account in negative sampling, so that overmuch negative sampling is not meaningful either.
- **Intermediate Region.** Unlike adjacent items, these items in the intermediate region are considered the positive-likeness hard items that can bring more information for model training. These intermediate items are a little far away from the central user u, and will degrade (or slightly improve) the recommendation performance when they are used as adjacent items to propagate information in the user-item graph. Compared with adjacent items, these intermediate items have a limited improvement in performance and sometimes even reduce the performance. In addition, propagating these intermediate items into the central node may lead to massive memory consumption since the nodes' size exponentially increases with the number of propagated layers. Thus, intermediate items should be sufficiently sampled as negatives to enhance the negative sampling.
- **Distant region.** A distant item refers to those which $p_d(v|u)$ is small. These distant items are usually far away

from the central user and present highly irrelevant properties. Commonly, the central user tend not to interact with these distant items so that the $p_d(v|u)$ (or $\mathbf{e}_u \cdot \mathbf{e}_v$) of these items is small. However, in recommendation, we usually care about top K items with the largest $\mathbf{e}_u \cdot \mathbf{e}_v$. A small $p_d(v|u)$ leads $\mathbf{e}_u \cdot \mathbf{e}_v$ to rapidly approach negative infinity, making overmuch negative sampling on them futile. In general, the distant items should be sampled less.

After clarifying the principle, there is still one unnegligible chasm before reaching a feasible algorithm: What is the criterion for separating three regions?

Separating Criterion. As shown in the three-region principle, we present the separating criterion to divide all items into three regions, which incorporates the structure of the user-item graph. Specifically, for a user u and subgraph as demonstrated in Figure 3, we leverage the Layer-wise Breadth First Search (LBFS) to traverse the subgraph and obtain the *personalized* items set. Here, we define the items within a u's $khop$-hop neighbors as the adjacent items and the items outside u's $khop$-hop neighbors as distant items. Thus, the items in u's $khop$-hop neighbors serve as the intermediate items. Notably, the traversed separating criterion involves the personalized regions for each user and concerns the structure of the user-item graph, demonstrating the diversity and personality. Here, we focus on the generation for intermediate items due to the intermediate region playing the decisive role for negative sampling. In light of the three-region principle, we argue that the region for negative sampling is *intermediate* rather than *global*. This principle will be verified in Section 5.5 (Deeper Study). The algorithm of LBFS is presented in Algorithm 1, where BFS used in Algorithm 1 is shown in Algorithm 2.

Moreover, we also provide the matrix form to speed up the construction of intermediate items. The $khop$-hop neighborhood $\mathbb{K}$ is defined as $\mathbb{K} = \mathbf{A}^{khop}$, where $\mathbf{A}$ is adjacent matrix of the user-item graph, and the nonzero cell of $\mathbb{K}$ represents the intermediate items. In this paper, we fix the $khop$ as $khop = 3$ and $khop = 5$ for Zhihu and Alibaba datasets respectively. Moreover, we also conduct an ablation study about the selection of $khop$ in Section 5.3.

Algorithm 1: The Layer-wise Breadth First Search (LBFS)

Input: The User-Item Graph $G = (\mathcal{U} + \mathcal{V}, \mathcal{O}^+)$, $khop$.
Output: The intermediate items region $\mathcal{R}_{med}$
for *each user* u **do**
  $hop\_num = 0$, $queue = [u]$.
  while *hop_num* $\leq khops$ **do**
    $y = BFS(G, queue)$.
    $queue = y$.
    hop_num += 1.
    if *hop_num* == $khop$ **then**
      $\mathcal{R}_{med} = y$.
    end
  end
end

Algorithm 2: The Breadth First Search (BFS)

Input: The User-Item Graph G, $queue$.
Output: The layer-wise node set y
for *each node in queue* **do**
  temp = $queue$.pop(0).
  nodes = G[temp].
  Add $nodes$ to y.
end

3.4 Discussion on The Three-Region Principle

As presented in the above descriptions, the three-region principle serves as a general principle to guide negative sampling. Compared with the global unobserved items, the items in the intermediate region provides more informative candidate negative items for model training. The three-region principle eliminates the interference of non-intermediate items and accurately samples candidate negative items from the intermediate region. In summary, the three-region principle is a general principle for negative sampling, which can be used for other existing negative sampling strategies by replacing the global unobserved item region with the intermediate region.

4 The RecNS Method

In this section, we illustrate the proposed negative sampling method RecNS, which is a general negative sampling method that can be directly plugged into existing graph-based recommendation models. RecNS designs two strategies for negative sampling: positive-assisted sampling (RecNS-O) and exposure-augmented sampling (RecNS-W), and merges these sampling strategies to generate the final negative item embeddings. The training flow of RecNS is illustrated in Algorithm 3. Different from prior arts [9], [16], [17], [19], [22], [23], where the region for negative sampling is global (u's unconnected edges in the user-item graph), the RecNS samples negative items from the intermediate region.

4.1 Positive-Assisted Sampling(RecNS-O)

To obtain hard negative items, we present the positive-assisted sampling strategy, in which the negative sampling is performed with the assistance of positive items. To be specific, for any given user u and the corresponding positive item v, the negative sampling distribution for the user u is determined by the user u and item v together, not only by the user u [16], [19], [23]. From the user u, the sampled negative item v_n reflects u's negative preference. Meanwhile, the negative one should possess a sufficient discrimination ability to distinguish between positive and negative items [24].

Inspired by the above studies, we introduce the idea of positive-assisted sampling by combining the positive item into the negative sampling distribution. We first follow the convention [16], [27] to select M candidate negative items from the intermediate region $\mathcal{R}_{med}$ constructed by the three-region principle to form the candidate items set $\mathcal{C}_u$, where M usually much smaller than the number of items in graph data. Next, we apply the inner product score to approximate the positive distribution and select the negative item from the candidate items set $\mathcal{C}_u$ with the estimated negative distribution p_n. Here, the negative distribution p_n is propor-

Algorithm 3: The training process with RecNS

Input: The intermediate region $\mathcal{R}_{med}$, Encoder E_θ, Number of Negatives k.
while *Early Stopping is not met* **do**
  Sample a mini-batch of positive pairs $\{(u, v)\}$.
  Initialize loss $\mathcal{L} = 0$.
  // Negative Sampling via RecNS.
  for *each* (u, v) *pair* **do**
    // Positive-Assisted Sampling.
    Get the candidate items set $\mathcal{C}_u$ from the intermediate region $\mathcal{R}_{med}$ by uniformly sampling M candidates.
    Get the sampled negative items set $\mathcal{P}_k$ by (7).
    // Exposure-Augmented Sampling.
    Get the self-amplified factor β by (9).
    Get the candidate exposed items set $\mathcal{M}_u$ from the exposed items set $\mathcal{E}_u$ by uniformly sampling M candidates.
    Get the exposed negative item v_n^e by (8).
    // RecNS.
    Merge the final negative embeddings by (10).
    $\mathcal{L} = \mathcal{L} + \max(0, \mathbf{e}_u^* \cdot \mathbf{e}_{v_n}^* - \mathbf{e}_u^* \cdot \mathbf{e}_v^* + \gamma)$
  end
  Update θ by descending the gradients $\nabla_\theta \mathcal{L}$.
end

tional to the positive distribution p_d, which is the derivation of MCNS [19]. Thus, we conduct the inner product score to estimate the positive-assisted sampling distribution:

$$p_n(v_n^p|(u,v)) = \frac{\sigma(\alpha(\mathbf{e}_u^* \cdot \mathbf{e}_{v_n^p}^*) + (1-\alpha)(\mathbf{e}_v^* \cdot \mathbf{e}_{v_n^p}^*))}{\sum_{v_i \in \mathcal{C}_u} \sigma(\alpha(\mathbf{e}_u^* \cdot \mathbf{e}_{v_i}^*) + (1-\alpha)(\mathbf{e}_v^* \cdot \mathbf{e}_{v_i}^*))} \tag{7}$$

where v_n^p is a sampled negative item from the distribution p_n, α is the assistance coefficient that balances the influence between the user u and the corresponding positive item v. Note that the assistance coefficient α is treated as a hyper-parameter to tune manually. We will empirically discuss the choice of distributions that assistance coefficient α obeys in Section 5.3, such as uniform distribution and Gaussian distribution.

In summary, the positive-assisted sampling enhances the negative sampling by incorporating positive information into negative sampling, which can help enforce the optimization process to exploit hard negative items for the decisive boundary. Worthly, we use the augmented hinge loss to optimize the parameters of the graph-based recommendation, where k negative items should be sampled from the positive-assisted sampling distribution p_n. These k negative items can form a sampled negative items set $\mathcal{P}_k = \{v_n^p\}$.

4.2 Exposure-Augmented Sampling(RecNS-W)

We present the exposure-augmented sampling strategy (RecNS-W) to sample negative items with the help of the exposure information. The E-commerce platform possesses the ability to collect exposure information, which is defined as whether the recommended item has been exposed to the user or not. Exposure information contains abundant information about the users' negative preferences. Intuitively, exposed but non-interacted items (abbreviated as exposed items) reflect the user's negative preferences compared with the global unobserved items. Besides, the false negative problem can also be alleviated with exposure information. Some works suggest that sampling negative items from the exposed items due to the exposed items possessing more negative properties. However, the strategy of sampling negative items from exposed items may face sampling bias since the exposed items themselves may be heavily biased, resulting in suboptimal performance [28]. Thus, we incorporate the exposure information into negative sampling and propose the exposure-augmented sampling to reduce the sampling bias and enhance the quality of negative items to lessen the false negative problem.

Specifically, for a user u and the corresponding exposed items set $\mathcal{E}_u = \{v_e\}$, we first uniformly sample M exposed items from the exposed items set $\mathcal{E}_u$ to form the candidate exposed items set $\mathcal{M}_u$. Next, we propose a self-amplified factor β to augment the influence of exposure information. Here we also utilize the inner product score to estimate the user u's preference over these M candidate exposed items and sample the exposed negative item v_n^e with the highest score, which is similar to the hard negative sampling strategy [16], [19]. Formally, the exposure-augmented sampling strategy is implemented as:

$$v_n^e = \underset{v_i \in \mathcal{M}_u}{\operatorname{argmax}} \ \sigma(\beta(\mathbf{e}_u^* \cdot \mathbf{e}_{v_i}^*)) \tag{8}$$

where $\cdot$ denotes inner product, and β is the proposed self-amplified factor. Recall that we only pick one exposed negative item v_n^e with the highest inner product score.

Here, we present the design of the self-amplified factor β. The main idea of the self-amplified factor is to leverage the sampled hard negative items in $\mathcal{C}_u$ to amend the inner product scores for exposed items. This design prefers to select exposed items that belong to candidate hard negative items as the candidate exposed negative items, which can correct the sampling bias problem caused by the exposed items themselves. To be specific, for a user u and any an exposed item v_e in the candidate exposed items set $\mathcal{M}_u$, if v_e belongs to the candidate items set $\mathcal{C}_u$, the magnitude of β would increase by one. Therefore, the magnitude of β is determined by the number of exposed items in $\mathcal{C}_u$. The self-amplified factor β is demonstrated as:

$$\beta = \begin{cases} 1, & \textbf{if } v_e \text{ not in } \mathcal{C}_u \\ \text{number of exposed items}, & \textbf{if } v_e \text{ in } \mathcal{C}_u \end{cases} \tag{9}$$

4.3 RecNS

In RecNS, the negative items are dependent on the above two sampling strategies: positive-assisted sampling and exposure-augmented sampling. Next, we introduce the method to combine two categories of sampling strategies. Essentially, the core idea of graph-based recommendation is iteratively propagating the embeddings in the user-item graph. Thus, we merge these sampling strategies in embedding space. The merge operation can be implemented as:

$$\mathbf{e}_{v_n}^* = \underset{v_n^p \in \mathcal{P}_k}{\operatorname{merge}}(\mathbf{e}_{v_n^e}^*, \mathbf{e}_{v_n^p}^*) \tag{10}$$

where v_n^p is sampled from the positive-assisted sampling distribution p_n and v_n^e is picked from the exposure-augmented sampling. $\mathcal{P}_k$ denotes the k negative items set which is generated from p_n. The merge operation merge$(\cdot)$ can be implemented without supernumerary parameters, in which the embeddings for v_n^p and v_n^e can be directly obtained from the GNNs-based encoder. Mathematically, the merge operation can be formalized as:

$$\text{merge}(\mathbf{e}_{v_n^e}, \mathbf{e}_{v_n^p}) = \frac{1}{k} \cdot \mathbf{e}_{v_n^e} + (1 - \frac{1}{k}) \cdot \mathbf{e}_{v_n^p} \tag{11}$$

where k is the number of negative items. Here, we don't need to involve supernumerary parameters to merge the negative embeddings in embedding space, which is a non-parametric merging process.

4.4 Discussion on RecNS

Universal Method. As presented in the above descriptions, the proposed negative sampling method RecNS is a universal method that can be plugged into graph-based recommendation models. Besides, RecNS uses the positive-assisted sampling to improve the quality of negative items, which is widely applied in a set of graph-based recommendation models. As for exposure-augmented sampling, the E-commerce platforms also can gather the exposure information to enhance negative sampling. In summary, the proposed RecNS is a universal method for negative sampling.

Embedding Mergence. Different from previous negative sampling methods that sample an item from the user-item graph, RecNS merges the negative embeddings between the positive-assisted sampling and the exposure-augmented sampling. Such a method merges the negative items in embedding space, which truly combines these two candidate negative items and promotes the quality of negative samples. In addition, the merge operation also provides a new method to utilize multiple negative items for the pairwise loss, that is, merging the embeddings of multiple negative items into the compound embedding.

4.5 Time Complexity

The computational complexity of RecNS comes from two parts. For positive-assisted sampling, the time cost is $O(Md)$, where M is the number of sampled candidates in $\mathcal{C}_u$, d represents embedding dimension. For exposure-augmented sampling, the complexity is attributed to the following two parts: (1) the computational complexity of the self-amplified factor β. The time for this part is $O(M)$. (2) The time cost for sampling one exposed negative item is $O(Md)$. The total computational complexity of exposure-augmented sampling is $O(M + Md)$ Thus, the time complexity of RecNS is $O(Md)$.

4.6 Three-Region Principle And RecNS

In general, the negative sampling depends on two parts from the perspectives of graph structure (iterative GNNs) and variance. The three-region principle is a general principle that guides the selection of regions for negative sampling; RecNS is a universal negative sampling method that satisfies the expected risk. The three-region principle provides an informative candidate negative item set while RecNS provides an efficient negative sampling distribution to sample negative items from candidate negative item set.

5 Experiments

To demonstrate the effectiveness and adaptiveness of the proposed RecNS, we conduct extensive experiments on two datasets with three representative graph-based recommendation models, such as PinSage, NGCF, and LightGCN. Next, we conduct a series of parameter analysis. Lastly, the deeper studies illustrate that why RecNS has superior negative sampling ability.

5.1 Experimental Settings

Datasets. We evaluate RecNS on two datasets: Zhihu and Alibaba datasets. For each user u, we randomly select 80% of user's interactions as the training set and use the next 10% of interactions as the validation set for hyperparameters tuning and early stopping. The remaining 10% interactions are used as the test set to evaluate the performance. Some statistics about these datasets are summarized in Table 1.

- **Zhihu** is a large QA website, where users click on interested articles to read. Here, we use a public dataset released in CCIR-2018 Challenge[1], which contains both article exposure information and user click information.
- **Alibaba dataset** collects user behaviors from the E-commerce platform Taobao. We sample a subset of data that contains users' historical interactions, including positive interactions and exposure information.

TABLE 1
Statistics of the two real-world datasets.

Dataset	# Users	# Items	# Edges	# Exposure	Density
Zhihu	16,015	44,175	3,284,734	3,796,551	0.00418
Alibaba	434,442	227,410	7,104,657	10,868,570	0.00006

Evaluation Metrics. We evaluate RecNS with three widely-used Top-K evaluation metrics, including Recall, NDCG, and HR, where K is set to 20. We report the average metrics for all users in the test set and compute the metrics by ranking all items that are not interacted by a user. Moreover, the efficient similarity search technique Faiss[2] is applied to extract search for efficient inner products.

GNNs-based Encoders. To verify the adaptiveness of RecNS to different genres of graph-based recommendation models, we utilize three commonly-used models as the trainable encoders for experiments.

- **PinSage** [1] utilizes the core idea of GraphSAGE [8] to learn item embeddings in web-scale graphs. In our experiments, we stack two aggregator layer to form information propogation paths for a user or an item, such as $u \leftarrow v \leftarrow u$ and $v \leftarrow u \leftarrow v$. In addition, the mean-aggregator is used in our experiments.
- **NGCF** [15] proposes a new recommendation framework named neural graph collaborative filtering, which exploits

1. https://www.biendata.xyz/competition/CCIR2018/
2. https://github.com/facebookresearch/faiss

the user-item graph structure by propagating embeddings on it. This leads to the expressive modeling of high-order connectivity in the user-item graph, effectively injecting the collaborative signal into the embedding process in an explicit manner.
- **LightGCN** [4] aims to simplify the design of GCN to make it more concise and appropriate for recommendation. LightGCN learns user and item embeddings by linearly propagating them on the user-item interaction graph and uses the weighted sum of embeddings learned at all layers as the final embedding.

Baselines. To demonstrate the effectiveness, we compare RecNS with different kinds of negative sampling methods, including includes static (UniNS and PopNS), heuristic (DNS and SimNS), adversarial (AdvNS), MCMC-based (MCNS), and SRNS strategies.

- **UniNS [9]:** Uniform negative sampling (UniNS) uses a uniform distribution to sample negative items. It is an independent negative sampling strategy that can be applied in various tasks, such as recommendation, information retrieval, and graph representation learning.
- **PopNS [29], [30]:** Popularity-based negative sampling (PopNS) samples negative items based on item popularity: $p(v|u) \propto pop(v)^{\alpha}$, where $pop(v)$ denotes the popularity of item v and α is a tuning parameter that affects the performance.
- **DNS [16]:** Dynamic negative sampling (DNS) is the state-of-the-art sampler, which adaptively samples negative items scored highest by the current recommendation model among some randomly selected items. Such negative ones is viewed as the hard negatives that can provide a large gradient for training.
- **SimNS [24]:** Similarity-based negative sampling strategy (SimNS) proposes a two-stage strategy to select informative negative items based on the distances between positive and negative items. This strategy takes into account the influence of positive items on negative sampling, which inspires us to incorporate the positive items into negative sampling.
- **AdvNS [17], [18]:** Adversarial negative sampling strategy (AdvNS), like IRGAN [17] and AdvIR [18], integrates the recommendation model into a generative adversarial network (GAN) where the generator performs as a negative sampler to select better negative items for confusing the discriminator (a.k.a recommendation model).
- **MCNS [19]:** Markov chain Monte Carlo negative sampling (MCNS) proposes an effective and scalable negative sampling strategy, approximating positive distribution with self-contrast approximation and accelerating negative sampling by Metropolis-Hastings.
- **SRNS [22]:** empirically observes that only a few instances are potentially important for model learning and samples negatives with high-variance to tackle the false negative problem.

Besides, we also evaluate RecNS in comparison with three typical negative sampling strategies integrating with exposure information, including ExpNS, MixedNS, and RNS-AS.

- **ExpNS:** ExpNS is firstly proposed to serve as a negative sampling strategy in MixedNS [31], which only selects negative items from the exposed item set. Such a strategy only focuses on exposure information but ignores the unobserved items, leading to suboptimal performance.
- **MixedNS [31]:** MixedNS integrates view signal (a.k.a exposure information) into negative sampling and designs a view-enhanced sampler for BPR. A user-oriented weighting strategy is considered during the learning process, which utilizes different sampling weight ω to sample exposed items and unobserved items.
- **RNS-AS [32]:** RNS-AS is applied to sample negative items integrating with exposure data, which combines the adversarial training and the feature matching together. In RNS-AS, the generator samples negative items from unobserved items and exposed items respectively, and utilizes the reward to update the training parameters.

Parameter Settings. We implement RecNS based on Tensorflow 1.14 and Python 3.7[3]. The embedding size is fixed to 64 for NGCF and LightGCN, while PinSage is set to 256. We use the Xavier method [33] to initialize embedding parameters for all encoders. The two aggregation layers with a fixed number of neighbors are applied for PinSage. The satisfactory performance of NGCF and LightGCN can be achieved when the number of propagation layers equals 3. We optimize all encoders with Adam [34] and use the default learning rate of 0.001 for PinSage and LightGCN, and keep the original learning rate of 0.0005 for NGCF. The L2 regularization coefficient is search in $\{1e^{-6}, 1e^{-5}, \cdots, 1e^{-2}, 1e^{-1}\}$, and set the optimum at $1e^{-3}$, $1e^{-4}$ and $1e^{-5}$ for PinSage, LightGCN and NGCF respectively. The early stopping strategy is conducted, where we stopped training if the Recall@20 on the validation set increased for 10 successive epochs. The margin γ is set to 0.1 in the augmented hinge loss for all encoders. Moreover, the number of negative items k is searched in [1, 5, 10, 15, 20], and the optimum is fixed as 15. Other parameters that were not mentioned but used in the encoders remain the default settings in the original papers.

5.2 Performance Comparison

Comparison with the State-of-the-Art Sampling Strategies. We summarize the detailed performance comparison among all negative sampling strategies on the Zhihu and Alibaba datasets in Table 2, where we highlight the results of best baselines (underlined) and RecNS (bold). Overall, RecNS brings about pronounced improvements on all datasets in terms of three evaluation metrics. To demonstrate the adaptiveness of RecNS, we apply RecNS into three representative graph-based recommendation models (PinSage, NGCF, and LightGCN). It is apparent that RecNS can significantly boost the recommendation performance. From Table 2, we have the following observations:

- RecNS consistently outperforms all baselines across three classical GNNs-based encoders. In specific, RecNS accomplishes remarkable improvements over the best baseline (SRNS) with the gains of Recall@20 by 8.28%, and 10.47% in Zhihu and Alibaba with PinSage encoder, respectively. These improvements are attributed to the following reasons: (1) Through the three-region principle, RecNS is

3. Codes are available at https://github.com/zyang-16/RecNS.

TABLE 2
Results of RecNS with SOTA negative sampling strategies. All the numbers in the table are percentage numbers with '%' omitted.

Methods	PinSage						NGCF						LightGCN					
	Zhihu			Alibaba			Zhihu			Alibaba			Zhihu			Alibaba		
	Recall	NDCG	HR	Recall	NDCG	HR	Recall	NDCG	HR	Recall	NDCG	HR	Recall	NDCG	HR	Recall	NDCG	HR
UniNS	2.42	2.72	34.55	3.02	1.34	4.58	3.78	4.31	46.95	3.71	1.72	5.31	4.32	4.92	50.47	5.43	2.45	7.60
PopNS	1.70	1.89	25.32	2.18	0.93	3.25	2.35	2.69	33.76	2.87	1.30	4.07	2.56	2.84	33.95	4.70	2.10	6.51
AdvNS	2.79	3.11	37.96	3.23	1.44	4.91	3.92	4.46	49.16	3.84	1.80	5.65	4.54	5.19	52.38	5.69	2.66	8.06
DNS	2.90	3.26	39.64	3.37	1.51	5.07	4.17	4.83	50.06	4.01	1.88	5.68	4.79	5.45	54.28	5.98	2.76	8.29
SimNS	2.94	3.28	39.51	3.40	1.52	5.07	4.09	4.67	49.71	3.84	1.79	5.48	4.49	5.21	52.70	5.91	2.76	8.21
MCNS	2.97	3.26	39.53	3.30	1.48	5.01	4.06	4.64	49.81	3.83	1.77	5.52	4.69	5.36	53.66	5.81	2.70	8.18
SRNS	3.26	3.74	42.87	3.63	1.65	5.29	4.24	4.90	50.66	4.32	2.01	6.04	4.83	5.45	55.14	6.34	2.99	8.80
RecNS	**3.53**	**3.93**	**44.57**	**4.01**	**1.82**	**6.04**	**4.42**	**5.00**	**51.89**	**4.58**	**2.17**	**6.44**	**4.91**	**5.55**	**55.37**	**6.86**	**3.22**	**9.51**

capable of exploring more informative candidate negative items from an intermediate region rather than the global unobserved item set. (2) The positive-assisted sampling strategy integrates positive items into negative sampling to balance the influence between the user and the positive item for negative sampling and samples more discriminative negative items to distinguish the positive and negative items for any given user. (3) With the augmentation of exposure information, RecNS can augment the negative sampling. Such exposure information provides the exposed items to reduce the sampling bias, further improving the quality of negative items.

- UniNS and PopNS are static and global negative sampling strategies and achieve poor performance on two datasets regardless of any graph-based encoders used. This indicates that these strategies usually cause non-optimal results. AdvNS beats the conventional strategies, like UniNS and PopNS. The reason is that the sampler learns to fit the user's preference distribution and adversarially generates "difficult" items that contribute more to training. Moreover, the recommender model gives a reward to guide the sampler to select negative items.
- The heuristic negative sampling strategies, comprising of DNS and SimNS, sample negative items based on the current model with a higher estimate score or higher ranking position. These sampling rules dynamically sample negative items based on current model along with the training process, boosting the recommendation performance.
- MCNS achieves higher performance than AdvNS in most cases. The reason is that the negative sampling distribution in MCNS is positively but sub-linearly correlated to positive distribution, which makes the deviation only negatively related to positive distribution, meaning that the inner products for high-probability negative items are estimated more accurately.
- SRNS outperforms other baselines over three classical graph-based models, which demonstrates that the importance of the false negative problem. SRNS selects items with high-variance as negatives and achieves efficient sampling of true negatives. Such sampling strategy enables us to tackle the risk of false negatives and improve the recommendation performance. Compared with SRNS, RecNS achieves a significant improvement on the Alibaba dataset. The reason is that RecNS utilizes exposure information to sample true negatives to alleviate the false negative problem, and applies the three-region principle to provide more informative candidates.

Comparison with Sampling Strategies Integrated with Exposure Information. We also conduct the experiments on three exposure-based negative sampling methods, including ExpNS, MixedNS, and RNS-AS. In Table 3, we summarize the performance comparison between RecNS and other exposure-based negative sampling methods. It can be seen through the results that RecNS is able to achieve better performance than baseline sampling methods. A few observations can be made as follows:

- It is obvious that sampling solely negative items from the exposed items (ExpNS) will lead to the worst recommendation performance. Some possible reasons are:(1) We can't determine the origin of the exposed items, which may belong to items that users dislike or that users ignore due to quick browsing. It is not a reliable negative signal by treating all exposed items as negatives, which may result in empirical bias. MixedNS needs to tune the sampling weight of ω between exposed items and global unobserved items and lacks flexibility. RNS-AS leverages exposure information and adversarial learning framework to select informative and real negative items adaptively and achieves better performance over other exposure information-based negative sampling strategies.
- RecNS proposes the exposure-augmented sampling strategy to incorporate the exposure information into negative sampling. Here, we use a self-amplified factor to augments the impact of exposed items and pick the exposed negative item with the highest inner product score. This design prefers to select more informative exposed items as the candidate exposed negative items. Besides, we merge the positive-assisted sampling and exposure-augmented sampling in the embedding space, which further enhances the quality of negative items.

5.3 Parameter Analysis

Impact of the $khop$-hop. To verify the impact of separating criterion $khop$ in the three-region principle, we conduct an ablation study by varying $khop$ in the range of $\{3, 5, 7\}$ to generate various intermediate items as the selected region for negative sampling. Note that we sample negative items for each user u, and thus the $khop$'s neighbors of u must belong to the item set. The experimental results are summarized in Table 4. We can obtain the following observations:

- It can be found that $khop$-3 achieves the highest performance on the Zhihu dataset, and the best performance on the Alibaba dataset is implemented at $khop$-5. The possible reason is that heavy propagation for a dense dataset makes its performance suffer from the degradation of negative preference. However, the degradation of negative

TABLE 3
Results of RecNS with exposure-based negative sampling strategies. All the numbers in the table are percentage numbers with '%' omitted.

Methods	Zhihu			Alibaba		
	Recall	NDCG	HR	Recall	NDCG	HR
PinSage+ExpNS	0.30	0.30	4.33	0.35	0.12	0.43
PinSage+MixedNS	2.09	2.40	31.64	2.79	1.24	4.27
PinSage+RNS-AS	2.98	3.34	39.92	3.52	1.59	5.21
PinSage+RecNS	**3.53**	**3.93**	**44.57**	**4.01**	**1.82**	**6.04**
NGCF+ExpNS	0.43	0.48	6.72	0.23	0.11	0.32
NGCF+MixedNS	3.12	3.67	42.45	2.75	1.28	3.90
NGCF+RNS-AS	3.67	4.27	46.45	2.82	1.32	4.03
NGCF+RecNS	**4.42**	**5.00**	**51.89**	**4.58**	**2.17**	**6.44**
LightGCN+ExpNS	0.76	0.84	11.35	0.95	0.43	1.33
LightGCN+MixedNS	3.67	4.25	46.65	3.97	1.82	5.53
LightGCN+RNS-AS	4.25	4.92	51.24	4.17	1.91	5.83
LightGCN+RecNS	**4.91**	**5.55**	**55.37**	**6.86**	**3.22**	**9.51**

preference for a sparse dataset would be postponed until the number of propagation modules reaches the optimum.

- Comparing different GNNs-based encoders, we can find that the three-region principle is applied to the existing graph-based recommendation models, boosting the quality of negative items. The selection of $khop$ shows consistency on various encoders, only demonstrating the dependence of the density of datasets. Thus, a simple and empirical separating criterion that $khop$ serves as 3 for a dense dataset and 5 for a sparse dataset, respectively.

TABLE 4
Impact of the separating criterion $khop$ that proposed in the three-region principle.

		Zhihu			Alibaba		
		Recall	NDCG	HR	Recall	NDCG	HR
PinSage	$khop$-3	3.53	3.93	44.57	3.69	1.67	5.57
	$khop$-5	3.26	3.60	42.48	4.01	1.82	6.04
	$khop$-7	3.20	3.59	42.42	3.86	1.75	5.81
NGCF	$khop$-3	4.42	5.00	51.89	4.24	1.98	6.04
	$khop$-5	4.18	4.74	50.07	4.58	2.17	6.44
	$khop$-7	4.21	4.76	49.95	3.68	1.73	5.38
LightGCN	$khop$-3	4.91	5.55	55.37	6.16	2.89	8.53
	$khop$-5	4.89	5.49	54.88	6.86	3.22	9.51
	$khop$-7	4.67	5.31	53.27	6.45	3.02	8.60

Impact of the Number of Propagation Modules. We conduct an experiment to analyze the impact of the number of propagation modules. We vary L in the range of $\{1, 2, 3, 4\}$ and demonstrate the results on NGCF and LightGCN in Table 5. In most cases, the recommendation performance is enhanced by increasing the number of propagation layers from 1 layer to 3 layer. Specifically, RecNS-2 can achieve a higher gain over RecNS-1, and the satisfactory performance usually performs at RecNS-3 in most cases. However, the performance of RecNS-4 drops compared with RecNS-3, the reason explained in LightGCN is that smoothing a node's embedding with higher-order neighbors will suffer from an over-smoothing issue. From the perspective of negative sampling, propagating higher-order neighbors that present the negative preference will degrade the performance.

TABLE 5
Impact of the number of propagation modules.

Dataset		Zhihu			Alibaba		
#Modules	Method	Recall	NDCG	HR	Recall	NDCG	HR
1 Module	NGCF	4.26	4.74	50.58	3.97	1.82	5.73
	LightGCN	4.56	5.29	53.19	4.41	2.08	6.15
2 Module	NGCF	4.34	4.87	51.52	4.11	1.93	6.00
	LightGCN	4.73	5.38	53.98	5.62	2.65	7.80
3 Module	NGCF	4.42	5.00	51.89	4.58	2.17	6.44
	LightGCN	4.91	5.55	55.37	6.86	3.22	9.51
4 Module	NGCF	3.94	4.47	48.24	4.26	2.00	6.05
	LightGCN	4.86	5.48	54.99	6.17	2.91	8.54

Impact of Number of Negative Items. To further investigate the superior performance brought by the augmented hinge loss. We visualize the convergence performance curve of PinSage+RecNS with the number of negative items k in Figure 4, and observe that increasing the k improves the performance of recommendation, verifying the effectiveness of the augmented hinge loss. It is observed that the Recall@20 performance increases with a large k at first, while performance begins to slightly decrease after reaching the optimum. Sampling more negative samples always reduces the risk, leading to an improvement in performance at first. However, performance begins to slightly decrease after reaching the optimum because extra bias is added to the objective by increasing k. In practice, although increasing k enhances the recommendation performance, the training time also increases. Thus, we set $k = 15$ to trade off the recommendation performance and training time.

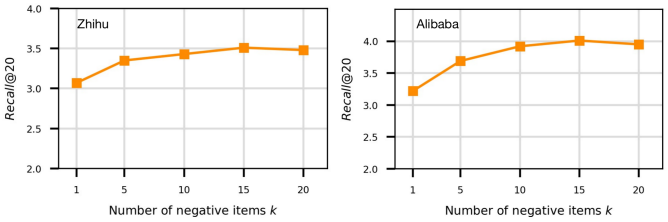


Fig. 4. Impact of number of negative items on the PinSage+RecNS. Results on NGCF and LightGCN show the same trend, which are omitted for space.

Impact of the assistance coefficience α. We conduct a series of experiments to investigate the obeyed distribution of assistance coefficience α. Here, we explore two classical distributions that α obeys, including Gaussian distribution $\mathrm{N}(\mu, \sigma^2)$ and Uniform distribution $\mathrm{U}(a, b)$. For Gaussian distribution, we use two types of distributions: $\mathrm{N}(0, 1)$ and $\mathrm{N}(0, 0.5)$. In addition, we fix the Uniform distribution with $a = 0$ and $b = 0.5, 1$ to study. The results with PinSage+RecNS are presented in Figure 5. The default setting in positive-assisted sampling is serve as uniform distribution $\mathrm{U}(0, 1)$, which achieves the best performance in all datasets. Compared with Uniform distribution, Gaussian distribution $\mathrm{N}(0, 0.5)$ reaches the comparable performance in the Alibaba dataset but performance slightly lower with $\mathrm{N}(0, 1)$. In summary, the obeyed uniform distribution of α can leads to better performance.

Impact of important hyperparameters. We investigate the impact of most important hyperparameters, margin γ, embedding dimension d, and the number of candidate items

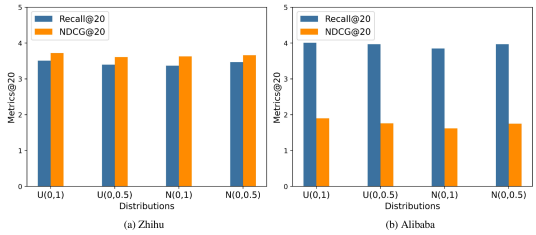


Fig. 5. Impact of the assistance coefficience α.

M. The results on the Zhihu dataset are reported in Figure 6. The augmented hinge loss is correlated to the margin γ and begins to come into effect when $\gamma > 0$. It is a disastrous consequence to set the margin $y \leq 0$, resulting in a zero loss problem. Figure 6 illustrates that the augmented hinge loss reaches its optimum at $\gamma \approx 0.1$. We search embedding dimensions in [64, 128, 256, 512, 1024, 2048] and keep the default setting ($d = 256$) in our experiments to trade-off performance and time consumption. Besides, we search the number of candidate items M in the range of [5, 10, 20, 30, 50] and keep the default setting ($M = 20$) in our experiments.

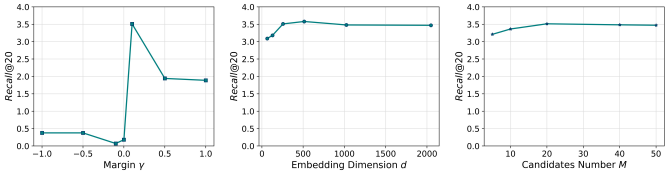


Fig. 6. Parameter analysis on PinSage+RecNS for the Zhihu dataset, including margin, embedding dimension, and candidates number.

5.4 Deeper Study

5.4.1 *Does The Three-Region Principle Improve Negative Sampling?*

RecNS conditions on the three-region principle. But does the principle help? To verify the impact of the three-region principle, we design one extended experiment that samples M candidates negative items from three regions, including intermediate region, distant region, and global unobserved region, termed as RecNS-Med, RecNS-Dis and RecNS-All respectively. The experimental results are presented in Figure 7.

As demonstrated in Figure 7, only sampling from *distant region* will lead to a catastrophic consequence, which illustrates that items in the distant region do not contribute positively to recommendation performance. Thus, we should avoid sampling negatives from *distant region*. Similarly, Sampling from the global unobserved region results in performance degradation since the global region cannot distinguish between intermediate and distant regions.

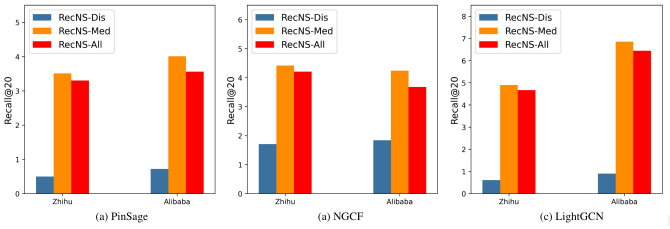


Fig. 7. The impact of the Three-Region Principle.

5.4.2 *Does The Positive-Assisted Sampling Enhance Negative Sampling?*

To answer this question, we design the experiment about sampling the hardest items in the training process. In practice, for any given user u, we approximate the hardest negative item v_n via sampling the highest inner product score between the user u and candidate negative items. Note that the hardest items are sampled by disabling the positive-assisted sampling strategy of RecNS, termed as RecNS$_{\text{w/o p-a}}$. We conduct the above experiment on Zhihu and Alibaba datasets. Figure 8 shows that sampling the hardest negative items will decrease recommendation performance, resulting in false negative instances issue [35]. Some works [1], [36] propose a simple workaround that only selects hard negative samples but avoids the hardest ones. Similar observations have also been discussed in previous related works [22], [37], suggesting sampling negative samples with both large scores and high variances to avoid false negative instances. We sample hard negative items with a balanced negative sampling distribution according to Equation (7), avoiding sampling the hardest ones.

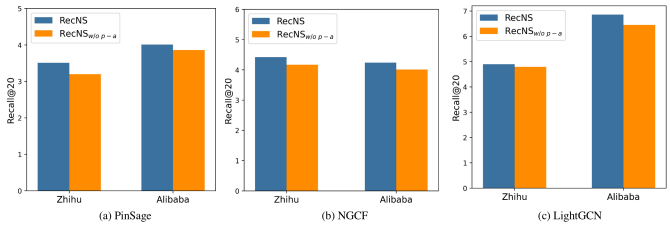


Fig. 8. The impact of the positive-assisted sampling strategy.

5.4.3 *Does The Exposure-Augmented Sampling Enhance Negative Sampling?*

We conduct two experiments to verify the impact of exposure-augmented sampling and represent the experimental results on Recall@20 with LightGCN encoder in Figure 9. The first experiment is to discard the exposure-augmented sampling of RecNS, termed as RecNS$_{\text{w/o e-a}}$. As shown in Figure 9, removing the exposure-augmented sampling decreases the recommendation performance, indicating the importance of exposure information. Integrating exposure information into negative sampling makes the negative sampler can sample true negatives to alleviate the false negative problem. Moreover, we also investigate the impact of the number of sampled exposed negatives on the Zhihu dataset with PinSage encoder and vary it in the range of $\{1, 5, 10, 15, 20\}$. The experimental result in Figure 9 demonstrates that sampling multiple exposed negative items can't improve recommendation performance because the exposed items suffer from heavy biases. Thus, we propose the self-amplified factor to enhance the influence of exposed items and pick the exposed negative item with the highest inner product score as a negative one.

6 Related Work

6.1 Graph-based Recommendation System

Recommendation systems have attracted a surge of attention because it is a core technology in many applications, such as E-commerce and social media. Conventional recommendation methods, e.g., collaborative filtering [38], matrix

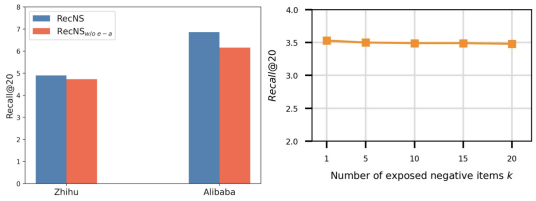


Fig. 9. The impact of the exposure-augmented sampling strategy and the impact of number of sampled exposed negative items.

factorization [39], [40], [41] cannot applied in web-scale recommendation system. With the rapid development of deep learning, Graph-based models can be used for recommendation systems. Yu et al. [42] proposed a linear model to aggregate user and item embeddings based on meta-path in heterogeneous information networks for recommendation. Pixie [43] uses pixie random walks to generate ranking scores by simulating random walks starting at the queried node. PinSage [1] develops a Graph Convolutional Network (GCN) to generate embeddings of users and items. Zhou et al. [3] proposed a graph embedding method via random walk with restart to capture asymmetric proximity. Wang et al. [2] proposed two aggregation methods to integrate the embeddings of items and the corresponding side information for recommendation. GC-MC [44] employs one convolutional layer to exploit the direct connections between users and items. NGCF [15] devises a new framework, which achieves the target by leveraging high-order connectivities in the user-item interaction graph. LightGCN [4] proposes two essential components, light graph convolution, and layer combination, to decrease the training difficulty.

6.2 Negative Sampling in Recommendation

Negative sampling is an effective method to slove one-class problem and speed up the training process in recommendation system [1], [12], [45], [46]. Here, we divide existing negative sampling methods into four categories and clarify the differences between RecNS with related works.

- **Static Sampler** samples negative items from a fixed negative sampling distribution. BPR [9] adopted uniform weights to sample negative items from missing data. Popularity-based negative sampling strategy [13], [30], [47] sampled negative items based on item popularity distribution.
- **Hard Negative Sampler** adaptively samples the hardest negative item based on the current recommendation models, which can accelerate the convergence compared with static sampler. Rendle et al. [23] applied an adaptive and context-dependent sampling distribution to oversample top ranked items. Zhang et al. [16] proposed a dynamic negative sampling (DNS) strategy that dynamically selects negative items from the ranked list produced by the current prediction model. CML [48] applied WARP loss [49] to a collaborative metric learning model, which uniformly sampled negative items within rejection. Tran et al. [24] proposed a 2-stage negative sampling strategy to find highly informative negative items. Instead of sampling negatives from the global unobserved items in the abovementioned negative sampling methods, RecNS utilizes the three-region principle to sample negatives from the intermediate region where can provides more informative candidate negatives for model training. RecNS is developed to serve the graph-based recommendation and is a merge sampling method which generates the final negative item embedding rather than a real negative item from the raw data. Moreover, RecNS introduces the idea of positive-assisted sampling by integrating the positive item into negative sampling distribution.
- **GAN-based Sampler** utilizes generative adversarial learning [50] to generate adversarial negative items. IRGAN [17] employs sampler to act as a generator and samples negatives to confuse the discriminative model. KBGAN [51] is an adversarial learning framework for knowledge graph embedding models by training two translation-based models. AdvIR [18] adds perturbation into adversarial sampling to make the model more robust.
- **Auxiliary-based Sampler** leverages auxiliary information (e.g. graph structure, social network, knowledge graph, user's exposure information) to sample informative negative instances. MCNS [19] proposes a theory to quantify the role of negative sampling in graph representation learning and utilizes Metropolis Hastings to accelerate negative sampling according to graph structure. SamWalker++ [52] extended the previous work [53], and proposed an efficient random walk-based sampling strategy along the pseudo-social network to draw informative training instances. KGPolicy [54] developed a reinforcement learning agent to explore high-quality negatives over item knowledge graphs. Besides, the additional view data is applied to sample negatives [31], [55]. RNS-AS [32] utilized adversarial learning and exposure data to sample negative items.

7 Conclusion

In this work, we study the problem of negative sampling in graph-based recommendation. Different from the previous works which ignore sampled regions for negative sampling, we propose the qualitative *Three-Region Principle* to guide negative sampling. This principle suggests that we should negatively sample more items at an intermediate region. Based on this principle, we present an effective negative sampling method called RecNS to sample hard negative items, which contains two sampling strategies: positive-assisted sampling and exposure-augmented sampling. Instead of sampling existing negative items, RecNS merge these two sampling strategies in embedding space to generate the final negative item embeddings. We conduct experiments on three representative graph-based recommendation models and the results suggest that RecNS can empower state-of-the-art graph-based recommendation models to achieve significant performance improvements over their default versions. In addition, the extensive experiments between RecNS and other commonly-used negative sampling methods are conducted, and the results show that the superiority of RecNS.

Acknowledgments

This work is supported by a research fund of the National Key R&D Program of China (2018YFB1402600), NSFC for Distinguished Young Scholar (61825602), NSFC (61836013), NSFC (61672313), and NSF under grants III-1763325, III-1909323, and SaTC-1930941.

REFERENCES

[1] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *KDD'18*. ACM, 2018, pp. 974–983.

[2] J. Wang, P. Huang, H. Zhao, Z. Zhang, B. Zhao, and D. L. Lee, "Billion-scale commodity embedding for e-commerce recommendation in alibaba," in *KDD'18*, 2018, pp. 839–848.

[3] C. Zhou, Y. Liu, X. Liu, Z. Liu, and J. Gao, "Scalable graph embedding for asymmetric proximity," in *AAAI'17*, 2017.

[4] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," *SIGIR'20*, 2020.

[5] S. Wu, W. Zhang, F. Sun, and B. Cui, "Graph neural networks in recommender systems: A survey," *arXiv preprint arXiv:2011.02260*, 2020.

[6] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *ICLR'17*, 2017.

[7] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *ICLR'18*, 2018.

[8] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NIPS'17*, 2017, pp. 1024–1034.

[9] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," *arXiv preprint arXiv:1205.2618*, 2012.

[10] Z. Xu, C. Chen, T. Lukasiewicz, Y. Miao, and X. Meng, "Tag-aware personalized recommendation using a deep-semantic similarity model with negative sampling," in *CIKM'16*, 2016, pp. 1921–1924.

[11] C. Li, Z. Liu, M. Wu, Y. Xu, P. Huang, H. Zhao, G. Kang, Q. Chen, W. Li, and D. L. Lee, "Multi-interest network with dynamic routing for recommendation at tmall," *CIKM'19*, 2019.

[12] J. Chen, C. Jiang, C. Wang, S. Zhou, Y. Feng, C. Chen, M. Ester, and X. He, "Cosam: An efficient collaborative adaptive sampler for recommendation," *ACM Transactions on Information Systems (TOIS)*, vol. 39, no. 3, pp. 1–24, 2021.

[13] H. Caselles-Dupré, F. Lesaint, and J. Royo-Letelier, "Word2vec applied to recommendation: Hyperparameters matter," in *RecSys'18*, 2018, pp. 352–356.

[14] T. T. Cai, J. Frankle, D. J. Schwab, and A. S. Morcos, "Are all negatives created equal in contrastive instance discrimination?" *arXiv preprint arXiv:2010.06682*, 2020.

[15] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *SIGIR'19*, 2019, pp. 165–174.

[16] W. Zhang, T. Chen, J. Wang, and Y. Yu, "Optimizing top-n collaborative filtering via dynamic negative item sampling," in *SIGIR'13*. ACM, 2013, pp. 785–788.

[17] J. Wang, L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, P. Zhang, and D. Zhang, "Irgan: A minimax game for unifying generative and discriminative information retrieval models," in *SIGIR'17*. ACM, 2017, pp. 515–524.

[18] D. H. Park and Y. Chang, "Adversarial sampling and training for semi-supervised information retrieval," in *WWW'19*, 2019, pp. 1443–1453.

[19] Z. Yang, M. Ding, C. Zhou, H. Yang, J. Zhou, and J. Tang, "Understanding negative sampling in graph representation learning," *KDD'20*, 2020.

[20] J. Chen, T. Ma, and C. Xiao, "Fastgcn: fast learning with graph convolutional networks via importance sampling," *ICLR'18*, 2018.

[21] W. Huang, T. Zhang, Y. Rong, and J. Huang, "Adaptive sampling towards fast graph representation learning," in *NIPS'18*, 2018, pp. 4558–4567.

[22] J. Ding, Y. Quan, Q. Yao, Y. Li, and D. Jin, "Simplify and robustify negative sampling for implicit collaborative filtering," *NIPS'20*, 2020.

[23] S. Rendle and C. Freudenthaler, "Improving pairwise learning for item recommendation from implicit feedback," pp. 273–282, 2014.

[24] V. Tran, R. Hennequin, J. Royoletelier, and M. Moussallam, "Improving collaborative metric learning with efficient negative sampling," pp. 1201–1204, 2019.

[25] N. E. Friedkin, *A structural theory of social influence*. Cambridge University Press, 2006, no. 13.

[26] A. Anagnostopoulos, R. Kumar, and M. Mahdian, "Influence and correlation in social networks," in *KDD'08*, 2008, pp. 7–15.

[27] J.-T. Huang, A. Sharma, S. Sun, L. Xia, D. Zhang, P. Pronin, J. Padmanabhan, G. Ottaviano, and L. Yang, "Embedding-based retrieval in facebook search," in *KDD'20*, 2020, pp. 2553–2561.

[28] R. Manmatha, C. Wu, A. J. Smola, and P. Krahenbuhl, "Sampling matters in deep embedding learning," pp. 2859–2867, 2017.

[29] X. He, H. Zhang, M. Kan, and T. Chua, "Fast matrix factorization for online recommendation with implicit feedback," pp. 549–558, 2016.

[30] H.-F. Yu, M. Bilenko, and C.-J. Lin, "Selection of negative samples for one-class matrix factorization," in *ICDM'17*. SIAM, 2017, pp. 363–371.

[31] J. Ding, G. Yu, X. He, Y. Li, and D. Jin, "Sampler design for bayesian personalized ranking by leveraging view data," *arXiv: Information Retrieval*, 2018.

[32] J. Ding, Y. Quan, X. He, Y. Li, and D. Jin, "Reinforced negative sampling for recommendation with exposure data," pp. 2230–2236, 2019.

[33] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *PMLR'10*, 2010, pp. 249–256.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR'15*, 2015.

[35] C.-Y. Chuang, J. Robinson, L. Yen-Chen, A. Torralba, and S. Jegelka, "Debiased contrastive learning," *arXiv preprint arXiv:2007.00224*, 2020.

[36] Y. Zhang, Q. Yao, Y. Shao, and L. Chen, "Nscaching: simple and efficient negative sampling for knowledge graph embedding," in *ICDE'19*. IEEE, 2019, pp. 614–625.

[37] H.-S. Chang, E. Learned-Miller, and A. McCallum, "Active bias: Training more accurate neural networks by emphasizing high variance samples," in *NIPS'17*, 2017, pp. 1002–1012.

[38] B. M. Sarwar, G. Karypis, J. A. Konstan, J. Riedl *et al.*, "Item-based collaborative filtering recommendation algorithms." *WWW'01*, vol. 1, pp. 285–295, 2001.

[39] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, no. 8, pp. 30–37, 2009.

[40] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *NIPS'08*, 2008, pp. 1257–1264.

[41] C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the β-divergence," *Neural computation*, vol. 23, no. 9, pp. 2421–2456, 2011.

[42] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han, "Personalized entity recommendation: A heterogeneous information network approach," in *WSDM'14*. ACM, 2014, pp. 283–292.

[43] C. Eksombatchai, P. Jindal, J. Z. Liu, Y. Liu, R. Sharma, C. Sugnet, M. Ulrich, and J. Leskovec, "Pixie: A system for recommending 3+ billion items to 200+ million users in real-time," in *WWW'18*. ACM, 2018, pp. 1775–1784.

[44] R. v. d. Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," *arXiv preprint arXiv:1706.02263*, 2017.

[45] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," in *ICDM'08*. IEEE, 2008, pp. 502–511.

[46] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," pp. 191–198, 2016.

[47] T. Chen, Y. Sun, Y. Shi, and L. Hong, "On sampling strategies for neural network-based collaborative filtering," in *KDD'17*, 2017, pp. 767–776.

[48] C. Hsieh, L. Yang, Y. Cui, T. Lin, S. Belongie, and D. Estrin, "Collaborative metric learning," pp. 193–201, 2017.

[49] J. Weston, S. Bengio, and N. Usunier, "Wsabie: Scaling up to large vocabulary image annotation," in *IJCAI'11*, 2011.

[50] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS'14*, 2014, pp. 2672–2680.

[51] L. Cai and W. Y. Wang, "Kbgan: Adversarial learning for knowledge graph embeddings," *arXiv preprint arXiv:1711.04071*, 2017.

[52] C. Wang, J. Chen, S. Zhou, Q. Shi, Y. Feng, and C. Chen, "Samwalker++: recommendation with informative sampling strategy," *arXiv preprint arXiv:2011.07734*, 2020.

[53] J. Chen, C. Wang, S. Zhou, Q. Shi, Y. Feng, and C. Chen, "Samwalker: Social recommendation with informative sampling strategy," in *WWW'19*, 2019, pp. 228–239.

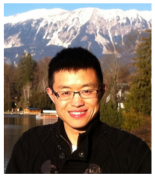
[54] W. Xiang, X. Yaokun, H. Xiangnan, C. Yixin, W. Meng, and C. Tat-Seng, "Reinforced negative sampling over knowledge graph for recommendation," *WWW'20*, pp. 99–109, 2020.

[55] B. Loni, R. Pagano, M. Larson, and A. Hanjalic, "Bayesian personalized ranking with multi-channel user feedback," in *RecSys'16*, 2016, pp. 361–364.

Zhen Yang is a PhD candidate in the Department of Computer Science and Technology, Tsinghua University. She got her master's degree from the Institute of Microelectronics, Tsinghua University. Her research interests include graph representation learning and graph-based recommendation.

Ming Ding is a PhD student in the Department of Computer Science and Technology in Tsinghua University. He also got his bachelor degree in Computer Science and Technology from Tsinghua University. His research interests include graph learning, natural language processing and cognitive artificial intelligence. He has published many papers on top conferences, such as KDD, ACL, IJCAI, etc.

Xu Zou is a PhD student in the Department of Computer Science and Technology, Tsinghua University. He got his bachelor degree from School of Aerospace, Tsinghua University. His research interests include computer vision, graph learning and natural language processing. He has published many papers on conferences such as CVPR and KDD.



Jie Tang is a Professor and the Associate Chair of the Department of Computer Science at Tsinghua University. He is a Fellow of the IEEE. His interests include artificial intelligence, data mining, social networks, and machine learning. He has published more than 200 research papers in major international journals and conferences. He was honored with the SIGKDD Test-of-Time Award.

Bin Xu is an associate professor in Department of Computer Science and Technology of Tsinghua University. He obtained his Ph.D., master and bachelor in Tsinghua University in 2006, 1998 and 1996 respectively. He became ACM Professional member in 2009 and IEEE member in 2007. His research interestes include semantic web, service computing and mobile computing.

Chang Zhou is working in Alibaba Group. He got his PhD degree from Peking University in 2017. His interests include representation learning, recommendation system, and graph computing. He has published over 20 over papers on conferences such as KDD, ACL, AAAI, etc.

Hongxia Yang is working as the Senior Staff Data Scientist and Director in Alibaba Group. She got her PhD degree in Statistics from Duke University in 2010. She has published over 30 over papers and held 9 filed/to be filed US patents and is serving as the associate editor for Applied Stochastic Models in Business and Industry.